УДК 681.3.06
DOI https://doi.org/10.32782/2663-5941/2024.3.1/28

Редько І.В.
Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»

Сущенко В.С.
Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»

ПРОБЛЕМИ ПОВНОТИ У КЛАСАХ ОБЧИСЛЮВАНИХ ФУНКЦІЙ І ПРЕДИКАТІВ НАД ІМЕНИМИ СТРУКТУРАМИ ДАНИХ

У статті розглянуті алгебраїчні характеристики класів обчислюваних функцій і предикатів над іменними структурами даних. У багатьох дослідженнях обґрунтована їхня роль як універсального засобу специфікації прагматично важливих для програмування структурних типів даних. Основний акцент зроблено на композиційній парадигмі програмування як методологічній основі розгляду методів побудови програм. Примітивні програмні алгебри (ППА) частково-рекурсивних функцій та частково-рекурсивнисх предикатів над іменними даними представляють об'єкт дослідження. Вибір саме таких структур даних обумовлений їх важливістю та популярністю, як у теоретичному, так і у прикладному програмуванні. Зокрема кортежні та реляційні структури даних грають важливу роль у дослідженнях, пов'язаних з повнотою мов запитів у базах даних. Точні визначення ППА наведені у роботах Редька В.Н., Буя Д.Б., Редька І.В. та інших авторів. Сигнатуру ППА складають операції суперпозиції, галуження та циклування, що уточнюють відомі управляючі структури більшості мов програмування і можуть застосовуватися при розробці прикладних телекомунікаційних програмно-апаратних систем. Особлива увага приділяється побудові ППА-характеристик (породжуючих множин та базисів) класів обчислюваних функцій та предикатів. Це складає предмет статті. Обчислюваність функції (предикату) над ефективно зліченими носіями вводиться як нумераційна обчислюваність, у сенсі робіт Катленда Н. «Обчислюваність. Вступ у теорію рекурсивних функцій», Єршова А.П. «Обчислюваність у довільних областях та базисах» та Єршова Ю.Л. «Теорія нумерацій». Представлена робота присвячена застосуванню алгебраїчного підходу та зокрема, методу ізоморфних специфікацій у дослідженні та розробці прикладних програмних систем. Разом з загальними результатами стосовно іменних даних та функцій, у роботі отримано описи класів кортежних та реляційних обчислюваних функцій та предикатів, визначено відповідні базисні набори функцій та предикатів у ППА.

Ключові слова: *алгебраїчна характеристика, композиційна парадигма, обчислювана функція, обчислюваний предикат, кортеж, реляція, програмна алгебра, ізоморфізм, Іт-базис.*

Постановка проблеми. Питання, пов'язані з розкриттям семантики мов програмування, відіграють ключову роль в теорії та практиці програмування. Дослідження класів обчислюваних функцій і предикатів над репрезентативними носіями, зокрема, визначення їх алгебраїчних характеристик, складає фундамент для технологізації процесів программотворення та розвитку засобів коректного програмування. Особливу роль в цих дослідженнях займають іменні структури даних. У [1–4] обґрунтована їх роль як універсального засобу специфікації прагматично важливих для програмування структурних типів даних.

Аналіз останніх досліджень і публікацій. Аналіз досліджень і публікацій, що стосуються відомих парадигм програмування, наприклад, функціонального, структурного, модульного, об'єктно-орієнтованого, композиційного програмування, програмних алгебр та програмології у цілому [1–3; 5–9] демонструє необхідність переходу у створенні прикладних інформаційних, телекомунікаційних систем, систем інтернету речей від вирішення окремих задач до підходів, способів та методів вирішення класів подібних задач з гарантованою коректністю отримуваних рішень та можливістю їх реалізації на різних синтаксичних програмних платформах.

Прагнення до першого було і залишається спонукальним мотивом створення більшості парадигм програмування. У тому чи іншому вигляді важливість цього знайшла своє відображення, зокрема, у Тьюрінговських лекціях Джона Бекуса, Едсгера Дейктра, Джона Мак-Карті, Роберта Флойда [9]. Визначні вчені, які внесли вагомий вклад у розбудову засад у тому числі і комп'ютерної науки та програмування у цілому, такі як Алонзо Чорч, Стівен Кліні, Гаскелл Каррі, Ніклаус Вірт, Алан Кей та ін. у своїй діяльності також керувались проблемами підвищення продуктивності та забезпечення коректності отримуваних рішень [10–13]. Реінжиніринг програмного забезпечення з лавиноподібним зростанням інвестицій у його розробку став самоочевидним трендом розвитку інформаційних технологій та програмування [14]. Віддаючи належне видатним досягненням, зробленим у цьому напрямку, тим не менше, необхідно звернути увагу на те, що всі вони з об'єктивних причин слабко інтегровані між собою [15]. Адже їх адекватна інтеграція потребувала розгляду ряду питань, для вирішення яких не було достатньої фактографії. Такими, зокрема, є питання об'єктивізації суб'єктивних впливів на вирішення задач, переходу від систем, орієнтованих на замкнуті у конкретиці рішення, до систем, орієнтованих на вирішення класів подібних задач.

Що ж до другого, то спрямованість композиційного підходу на фактологічні дослідження притаманних програмуванню генетичних структур дозволяє не тільки коректно ставити, але й ефективно вирішувати задачі синтезу прикладних інформаційних та телекомунікаційних систем. На відміну від більшості традиційних підходів, що орієнтовані на синтаксичну нотацію результатів свідомого чи несвідомого застосування таких генетичних структур.

Постановка завдання. Метою даної роботи є визначення алгебраїчних характеристик класів обчислюваних функцій і предикатів над рядом іменних структур даних. Класи частково-рекурсивних функцій та предикатів над іменними носіями складають об'єкт дослідження. Предметом роботи є алгебраїчні характеристики класів частково-рекурсивних функцій та предикатів над іменними структурами даних типу кортежів та реляцій у примітивних програмних алгебрах (див. [16; 17] та бібліографію).

Виклад основного матеріалу. *Загальні положення та необхідні позначення.* Носій ППА складають n-арні функції та n-арні предикати (далі – функції та предикати) ($n = 1,2,\cdots$). Сигнатуру ППА (позначаєм тут як **Ω**) складають операції суперпозиції, гілкування та циклування, як уточнення основних методів побудови програм [16; 17].

Розглянемо деяку злічену множину D, що трактується як абстрактний тип даних, і для будь-якого натурального $k > 0$ розглянемо класи Φ^k обчислюваних (частково-рекурсивних) k-арних функцій та предикатів вигляду $D^k \to D$ та $D \to \{T,F\}$ відповідно, та клас

$$\Phi \equiv \bigcup_{k=0} \Phi^k, k = 1,2,3$$

багатомісних частково-рекурсивних (обчислюваних) функцій та предикатів на D. Під функціями (предикатами) на D, або D-функціями (D-предикатами), будемо розуміти функції (предикати) з Φ. Обчислюваність на D вводиться як нумераційна обчислюваність [18; 19].

Домовимося через $A_D^{чр}$ позначати ППА, носій якої складають частково-рекурсивні функції (*чр*-функції) і частково-рекурсивні предикати (*чр*-предикати) на D, тобто $A_D^{чр} \equiv \Phi, \Omega$. Породжуючу множину алгебри $A_D^{чр}$ назвемо її повною системою (ПС) і позначатимемо $\sigma_D^{чр}$. Повну систему ППА називатимемо її I_m^n-базисом, якщо будь-яка її підсистема, отримана видаленням з неї будь-якого предиката чи будь-якої функції, відмінної від селекторної, вже не буде повною.

Зафіксуємо V та W – непорожні зліченні множини елементів, що трактуються відповідно як множини імен та значень (денотатів). У загальному випадку допускається, що деякі імена можуть виступати в ролі значень і навпаки, тобто можливо, що $V \cap W \neq \varnothing$. Елементи множин V та W називатимемо простими даними. Розглянемо тип даних D, породжений наступною індуктивною процедурою. Домовимось $\mathrm{W} \subseteq \mathrm{D}$. Будь-які інші елементи з D породжуються такими правилами:

1) якщо імена $v_1, \ldots v_k$ з V попарно різні та є $d_1, \ldots, d_k \in D$ то $(\nu_1, d_1), \ldots, (\nu_k, d_k) \in D, (k \geq 0)$;

2) якщо $d_1, \ldots, d_s \in D$ то $d_1, \ldots, d_s \in D, (s \geq 0)$.

Так введений тип D називатимемо типом іменних даних [1; 2]. З огляду на те, що D залежить від параметрів V та W, в разі необхідності явного підкреслення цієї залежності для типу іменних даних використовуватимемо також позначення $D^{(V,W)}$, а елементи з D – називатимемо (V,W)-іменними даними.

Вже поверхневого розгляду типу іменних даних D достатньо, щоб зрозуміти, що він базується на трьох утворюючих – абстрактному типі W та породжуваних однократним застосуванням правила *1* примітивно-іменному та правилом *2* – примітивно-множинному типах даних $PrN(V,W)$ та PrS^W, відповідно. Під примітивно-іменним даним (для стислості використовуватимемо також терміни (V,W)-іменна множина, іменна множина) розумітимемо скінченне функціональне бінарне відношення між множинами імен V та значень W. Під примітивно-множинним даним (також W-множина, множина) розумітимемо скінченну підмножину з W. Тобто, $\mathrm{PrN}(\mathrm{V},\mathrm{W}) \subset 2^{\mathrm{V} \times W}$ та $\mathrm{PrS}^{\mathrm{W}} \equiv 2^{\mathrm{W}}$, де $\equiv$ означає тотожність за визначенням, а запис 2^A для деякої заданої множини A – множину всіх скінченних підмножин множини A.

Структурологічне збагачення іменних даних. Введений тип іменних даних D структурно збагачує абстрактне розуміння даних і дозволяє згідно до принципу обумовленості [1; 2], розглянути дані у контексті породжуючих їх процесів абстрагування, омножування та іменування у їхньому взаємодоповненні. З цього випливає, що іменний тип даних відкритий для широкого спектру подальших прагматично обумовлених збагачень, що ґрунтуються на взаємодоповненні вищезгаданих трьох складових – W, $PrN(V,W)$ та PrS^W. Низка таких предметних збагачень, які базуються, як на виборі параметрів V та W, так і на обмеженнях в застосуванні правил *1* і *2*, буде розглянута нижче. Усі вони реалізують вже не абстрактну точку зору на дані з D, а розглядають їх у контексті їх породжень.

За ознакою покроковості застосування правил *1* і *2* розглянемо такі збагачення:

$1^{(V,W)}$ – (V,W) – тип іменних даних, побудований лише за допомогою правила *1*:

$$1^{(V,W)} \equiv \bigcup_{i=1}^{\infty}\left(PrN^{(V,W)}\right)^i,$$

де

$$\left(PrN^{(V,W)}\right)^1 \equiv PrN^{(V,W)},$$

$$\left(PrN^{(V,W)}\right)^2 \equiv PrN^{\left(V,\left(PrN^{(V,W)}\right)^1\right)},\cdots,$$

$$\left(PrN^{(V,W)}\right)^i \equiv PrN^{\left(V,\left(PrN^{(V,W)}\right)^{i-1}\right)},\cdots;$$

$PrN^{(V,D)}$ – тип іменних даних, побудова кожного з елементів якого завершується правилом *1*;

$2^{(W)}$ – тип іменних даних, побудований лише за допомогою правила *2*:

$$2^{(W)} \equiv \bigcup_{i=1}^{\infty}\left(PrS^{(W)}\right)^i,$$

де

$$\left(PrS^{(W)}\right)^1 \equiv PrS^{(W)},$$

$$\left(PrS^{(W)}\right)^2 \equiv PrS^{\left(\left(PrS^{(W)}\right)^1\right)},$$

$$\left(PrS^{(W)}\right)^i \equiv PrS^{\left(\left(PrS^{(W)}\right)^{i-1}\right)},\cdots;$$

$PrS^{(D)}$ – тип іменних даних, побудова кожного з елементів якого завершується правилом *2*;

$PrN^{\left(V,PrS^{(D)}\right)}$ і $PrN^{\left(V,PrS^{(W)}\right)}$-іменні дані типу $\left(V,PrS^{(D)}\right)$-іменних та $\left(V,PrS^{(W)}\right)$-іменних множин, відповідно;

$PrS^{\left(PrN^{(V,D)}\right)}$ і $PrS^{\left(PrN^{(V,W)}\right)}$-іменні дані типу $PrN^{(V,D)}$-множин та $PrN^{(V,D)}$-множин, відповідно;

Останні збагачення $PrN^{\left(V,PrS^{(W)}\right)}$, $PrN^{\left(V,PrS^{(D)}\right)}$, $PrS^{\left(PrN^{(V,W)}\right)}$, $PrS^{\left(PrN^{(V,D)}\right)}$ є платформою подальших прагматично обумовлених типізацій іменних даних D. Для їх здійснення буде потрібно ввести кілька додаткових визначень і для зручності викладення домовитися про кілька корисних позначень.

Під схемою іменної множини K будемо розуміти скінчену множину імен, позначувану $Sh(K)$, що представляє собою проекцію K за першою компонентою, тобто $Sh(K) = pr_i(K)$, де pr_i – функція проекції за i-ю компонентою m-арного відношення $(1 \le i \le m)$ [1; 2; 15; 16]. При цьому, іменну множину K будемо називати для скорочення $sh(I)$-іменною множиною або іменною множиною зі схемою $sh(I)$ і позначати $K^{sh(I)}$. У розгорнутій формі будемо представляти $\{v_1,\cdots,v_n\}$-іменні множини так: $K^{\{v_1,\cdots,v_n\}} \equiv \{(v_1,d_1),\cdots,(v_n,d_n)\}, v_i \in \mathrm{V}, d_i \in D, \mathrm{i}=1\cdots n$. Іменні множини з однаковими схемами домовимося називати односхемними іменними множинами.

Множину всіх (V,D)-іменних множин зі схемою $\{v_1,\cdots,v_n\}$ позначимо $\mathrm{Pr}N^{(V,D)}\left[\{v_1,\cdots,v_n\}\right]$, а множину всіх примітивно-іменних даних з тією ж схемою – $\mathrm{Pr}N^{(V,W)}\left[\{v_1,\cdots,v_n\}\right]$. При цьому, неважко переконатись, що

$$PrN^{(V,W)}\left[\{\mathrm{v}_1,\cdots,\mathrm{v}_n\}\right] \subset PrN^{(V,D)}\left[\{\mathrm{v}_1,\cdots,\mathrm{v}_n\}\right].$$

Маючи на увазі, що $K^{\varnothing} = \varnothing$ та $PrN^{(V,D)}[\varnothing] = \varnothing$, отримаємо

$$PrN^{(V,D)} = \bigcup_{V'=2^V} PrN^{(V,D)}\left[V'\right]$$

і зокрема,

$$PrN^{(V,W)} = \bigcup_{V'=2^V} PrN^{(V,W)}\left[V'\right].$$

Множини всіх $PrN^{(V,D)}\left[\{\mathrm{v}_1,\cdots,\mathrm{v}_n\}\right]$-множин та $PrN^{(V,W)}\left[\{\mathrm{v}_1,\cdots,\mathrm{v}_n\}\right]$-множин позначимо $\mathrm{Pr}S^{(D)}\left[\{v_1,\cdots,v_n\}\right]$ і $\mathrm{Pr}S^{(W)}\left[\{v_1,\cdots,v_n\}\right]$. Зрозуміло, що

$$PrS^{(D)}\left[\{\mathrm{v}_1,\cdots,\mathrm{v}_n\}\right] = 2^{PrN(V,D)\left[\{\mathrm{v}1,\cdots,\mathrm{v}n\}\right]},$$

а також

$$PrS^{(W)}\left[\{\mathrm{v}_1,\cdots,\mathrm{v}_n\}\right] = 2^{PrN^{(V,W)}\left[\{\mathrm{v}_1,\cdots,\mathrm{v}_n\}\right]}.$$

Звідси, за аналогією з попереднім,

$$\mathrm{Pr}S^{(W)}\left[\{v_1,\cdots,v_n\}\right] \subset \mathrm{Pr}S^{(D)}\left[\{v_1,\cdots,v_n\}\right]$$

$$PrS^{(D)} = \bigcup_{V'\in 2^V} PrS^{(D)}\left[V'\right],$$

та

$$PrS^{(W)} = \bigcup_{V'\in 2^V} PrS^{(W)}\left[V'\right].$$

Важливою особливістю типу $PrS^{(D)}\left[\{v_1,\cdots,v_n\}\right]$ і зокрема, $PrS^{(W)}\left[\{v_1,\cdots,v_n\}\right]$, є те, що їхніми елементами є множини односхемних *D*- та *W*-іменних множин. У зв'язку з цим, $PrS^{(D)}\left[\{v_1,\cdots,v_n\}\right]$ та $PrS^{(W)}\left[\{v_1,\cdots,v_n\}\right]$ можуть бути базою для експлікації значимих для комп'ютерної науки яких завгодно складно організованих структурних типів даних, таких як таблиця, реляція, масив, список, черга і т. п. Усі вони є видами згаданих вище родових типів, які ми називатимемо, відповідно до традицій, $\{v_1,\cdots,v_n\}$-арними *D*- та *W*-множинами, або, якщо немає необхідності вказувати явно схему іменних множин їх складових – поліарними *D*- та *W*-множинами. Самі ж типи $PrS^{(D)}\left[\{v_1,\cdots,v_n\}\right]$ і $PrS^{(W)}\left[\{v_1,\cdots,v_n\}\right]$ – *D*- і *W*-поліарними типами даних або просто поліарними типами даних, якщо це не призведе до непорозумінь. Експлікація значущих для програмування структурних типів даних може бути проведена за допомогою подальшої типізації поліарних типів даних.

Проте ці типізації матимуть вже не структурологічний, а предметний характер, пов'язаний з конкретизаціями множин *V* та *W*. Крім того, новостворені структурні типи даних слід збагатити притаманними їм функціональними структурами.

Предметна конкретизація поліарних типів даних. Як перший крок предметної конкретизації зафіксуємо $V = W = N$. Відзначимо, що через зліченність множин *V* та *W* такий перехід, з одного боку, принципово не обмежить загальність розглядів, а з іншого боку, суттєво спростить викладення отриманих результатів. При цьому, дана конкретизація суттєво збагачує розгляди, підтримуючи виділення нових, прагматично обумовлених типів даних. Деякі з них є збагаченнями «по формі» і зводяться до результатів простого перепозначення імен і денотатів. Збагачення «по суті», зі свого боку, зводяться до врахування послідовнісної специфіки структур даних, характерних для сучасного програмування. До перших відносяться типи: $PrN^{\left(N,D^{(N,N)}\right)}\left[\{v_1,\cdots,м_n\}\right]$, $PrN^{(N,N)}\left[\{v_1,\cdots,v_n\}\right]$, $PrS^{\left(D^{(N,N)}\right)}\left[\{v_1,\cdots,v_n\}\right]$ та $PrS^{(N,N)}\left[\{v_1,\cdots,v_n\}\right]\in 2^N$. Другі є їх суттєвими конкретизаціями «за наборами імен» – результатами переходу від довільної схеми іменування $\{v_1,\cdots,v_n\}$ до стандартної, що ґрунтується на «щільній» послідовній схемі – $\{1,\cdots,\mathrm{n}\}\big|_{n\in N}$: $PrN^{\left(N,D^{(N,N)}\right)}\left[\{1,2,\cdots,n\}\right]$, $PrN^{(N,N)}\left[\{1,2,\cdots,n\}\right]$, $PrS^{\left(D^{(N,N)}\right)}\left[\{1,2,\cdots,n\}\right]$ та $PrS^{(N)}\left[\{1,2,\cdots,n\}\right]$

Так наприклад, $PrN^{(N,N)}\left[\{1,2,\cdots,n\}\right]$, представляє собою іменну експлікацію широко використовуваного у програмуванні кортежного типу: наприклад, експлікацією кортежу $\langle 3,5,7\rangle$ є іменна множина:

$$\{(1,3),(2,5),(3,7)\}\in PrN^{(N,N)}\left[\{1,2,3\}\right].$$

При цьому дана експлікація не просто відображає кортежну специфіку експлікованої сутності, але надає концептуально єдине розуміння його трактування її в більшості мов програмування.

$\mathrm{PrS}^{(\mathrm{N})}\left[\{1,2,\cdots,\mathrm{n}\}\right]$ експлікує тип скінченних *n*-арних відношень, які складають основу широко використовуваного у комп'ютерній науці реляційного типу:

1	2	3
5	8	4
2	7	2

$$\Leftrightarrow \{\{(1,5),(2,8),(3,4)\},\{(1,2),(2,7),(3,2)\}\}\in PrS^{(N,N)}\left[\{1,2,3\}\right]$$

Іменні експлікації структур даних можна продовжити за необхідності. Важливо відзначити, що всі вони базуватимуться на концептуально єдиній основі – іменному даному. Тепер розглянемо алгебраїчні характеристики класів обчислюваних функцій над окремими репрезентативними носіями, в рамках відповідних примітивних програмних алгебр (ППА).

Алгебраїчні характеристики класів частково-рекурсивних функцій та предикатів над іменними даними

З зазначеного випливає можливість використання іменної моделі даних як концептуально єдиної платформи експлікації структурних типів даних. Тут розглянемо алгебраїчні характеристики класів обчислюваних функцій над кортежами та реляціями. У якості платформи розгляду оберемо примітивні програмні алгебри (ППА). У якості методу використаємо метод ізоморфних специфікацій [2; 16].

Нехай $K\equiv\bigcup\limits_{i\in N}PrN^{(N,N)}\left[\{1,2,\cdots,i\}\right]$ – кортежний, а $R\equiv\bigcup\limits_{i\in N}PrN^{(N,N)}\left[\{1,2,\cdots,i\}\right]$ – реляційний носії даних, відповідно, де $PrN^{(N,N)}[0]\equiv\rangle$ – т. з. «пустий» кортеж, що не містить елементів, а $PrS(N)[0]\equiv\Delta$ – «пуста» реляція» як пуста множина кортежів. Домовимось малими літерами латинського алфавіту k,l,m, можливо із індексами, якщо не оговорене або з контексту їх використання не слідує інше, будемо позначати елементи множини *K*, літерами q,r,s,t, можливо з індексами – елементи множини *R*.

Зазначимо, що з ефективної зліченості множин K та R безпосередньо випливає їх ефективна впорядкованість. Звернемося до ППА $A_K^{чр} = \Phi_K$, частково-рекурсивних функцій (*чр*-функцій) та частково-рекурсивних предикатів (*чр*-предикатів), як платформи уточнень кортежних маніпуляцій.

Відомо, що набір

$$\sigma_K \equiv \left\{\Phi, S, C_0, \circ, =_K, I_m^n \big|_{(m,n\in N, m\le n)}\right\}$$

представляє собою I_m^n-базис алгебри $A_K^{чр}$ [8]. Тут Φ, S, C_0 – унарні, $\circ$ – бінарна, I_m^n – n-арні ($n,m\in N$) *чр*-функції та $=_K$ – бінарний *чр*-предикат такі, що:

$$\Pi:\Pi\left(\left\{(1,a_1),(2,a_2),\cdots,(s,a_s)\right\}\right)\equiv$$
$$\equiv\left\{(1,a_2),(2,a_3),\cdots,(s,a_{s-1})\right\}\big|_{s,a_i\in N, i=\overline{1,s}}$$

– видалення першого елементу кортежу;

$$S:S\left(\left\{(1,a_1),(2,a_2),\cdots,(s,a_s)\right\}\right)\equiv$$
$$\equiv\left\{(1,a_1+1),(2,a_2),\cdots,(s,a_s)\right\}\big|_{s,a_i\in N, i=\overline{1,s}}$$

– збільшення першого елементу кортежу на одиницю;

$$C_0:C_0\left(\left\{(1,a_1),(2,a_2),\cdots,(s,a_s)\right\}\right)\equiv\left\{(1,0)\right\}\big|_{s,a_i\in N, i=\overline{1,s}}$$

– встановлення константи $\left\{(1,0)\right\}$;

$$\circ:\circ\begin{pmatrix}\left\{(1,a_1),(2,a_2),\cdots,(s,a_s)\right\}\big|_{s,a_i\in N,\, i=\overline{1,s}}, \\ \left\{(1,b_1),(2,b_2),\cdots,(k,b_k)\right\}\big|_{k,a_j\in N,\, j=\overline{1,k}}\end{pmatrix}\equiv$$
$$\equiv\{((1,a_1),\cdots,(s,a_s),(s+1,b_1),\cdots,(s+k,b_k)\}\big|_{s,a_i,k,a_j\in N,\, i=\overline{1,s} j=\overline{1,k}}$$

– конкатенація двох кортежів;

$$I_m^n:I_m^n\left(A_1,\cdots,A_n\right)=A_m\big|_{m,n\in N, m\le n},$$

де

$$A_i=\left\{(1,a_1^i),\cdots,(p_i,a_{p_i}^i)\right\}\big|_{i=\overline{1,n}, p_i\in N}$$

– селекторна функція [8; 14];

$$=_k:=_k\left(\left\{(1,a_1),\cdots,(s,a_s)\right\}\big|_{s,a_s\in N, i=\overline{1,s}},\left\{(1,b_1),\cdots,(k,b_k)\right\}\right)\big|_{k,b_k\in N, j=\overline{1,k}}=$$
$$=\begin{cases} T, якщо(s=k)\wedge(i=j)\rightarrow(a_i=b_j) \\ F, інакше\end{cases}$$

– предикат рівності кортежів.

Спираючись на результат про I_m^n-базис алгебри $A_K^{чр}$ та на ефективну зліченість множин K та R, співставляючи кожній *чр*-функції на R *чр*-функцію на K, яка представляє її у відображенні:

$$H:K\rightarrow R:H(\Lambda)=\Delta, H(\left\{(1,a_1),\cdots,(m,a_m)\right\})\big|_{m,a_i\in N, i=\overline{1,m}}\equiv$$
$$\equiv\left\{\left\{(1,a_1),\cdots,(m,a_m)\right\}\right\}\big|_{m,a_i\in N, i=\overline{1,m}}$$

та діючи аналогічно для *чр*-предикатів, будуємо бієкцію $\theta_H: R\rightarrow K$-ізоморфізм алгебри $A_R^{чр}$ на алгебру $A_K^{чр}$. Для цього, визначимо кодуюче та декодуюче відображення $\phi: R\rightarrow K$ та $\vartheta: K\rightarrow R$ і змоделюємо їх у вигляді реляційних *чр*-функцій $\psi: R\rightarrow H(K)$ та $\chi: H(K)\rightarrow R$, де $H(K)\subset R$ – образ множини K у відображенні H. Зауважимо, що зліченість множин K та R обумовлює впорядкованість будь-якої реляції як множини кортежів. Будемо вважати, що кортежі впорядковані у відповідності з їх номерами у деякій зафіксованій ефективній нумерації множини K. Тоді кодуюче та декодуюче відображення ϕ та ϑ можуть бути задані так:

$$\phi(\Delta)=\Lambda,$$
$$\phi(\left\{\left\{(1,a_1^i),\cdots(j,a_j^i)\cdots,(k,a_k^i)\right\}\right\}\big|_{i=\overline{1,n}, j=\overline{1,k}, a_j^i, k,n\in N}=$$
$$=\left\{(1,k)\right\}\circ\left\{(2,a_1^{i_1}),\cdots,(k+1,a_k^{i_1})\right\}\circ\cdots$$
$$\cdots\circ\left\{(s\cdot k+2,a^{i_{s\cdot k}}),\cdots,((s+1)\cdot k,a_k^{i_{s\cdot k}})\right\}\circ$$
$$\circ\left\{((s+1)\cdot k+1,a^{i_{s\cdot k+1}}),\cdots,(n\cdot k,a_k^{i_{n\cdot k}})\right\}\big|_{s=\overline{1,n}}$$

та

$$\vartheta(\Lambda)=\Delta,$$
$$\vartheta\left(\left\{(1,k),(2,a_1),\cdots,(s\cdot k,a_{s\cdot k})\right\}\big|_{k,s,a_i\in N, i=\overline{1,s\cdot k}}\right)=$$
$$=\begin{Bmatrix}\left\{(1,a_1),\cdots,(k,a_k)\right\},\left\{(1,a_{k+1}),\cdots,(k,a_{2\cdot k})\right\},\cdots \\ \cdots,\left\{(1,a_{(s-1)\cdot k}),\cdots,(k,a_{s\cdot k})\right\}\end{Bmatrix}$$

Використавши, з метою моделювання кортежних функцій реляційними, введене вище відображення H отримуємо $\psi=\phi\cdot H$. Тут $\phi\cdot H$ розуміється як традиційне множення функцій, тобто $\phi\cdot H: R\rightarrow R$ така, що $\phi\cdot H(Q)=H(\psi(Q))\big|_{Q\in R}$. Відповідно, функцію $\chi: R\rightarrow R$ можна представити як деяке розширення функції ψ^{-1} на всю множину R. Очевидно, що так уведені функції ψ та χ є реляційними *чр*-функціями, що представляють у відображенні H кодуюче та декодуюче відображення ϕ та ϑ.

Розглянемо наступні реляційні *чр*-функції та *чр*-предикати:

$$\Pi_R:\Pi_R\left(\begin{Bmatrix}\left\{(1,a_1^1),(2,a_2^1),\cdots,(s,a_s^1)\right\},\cdots \\ \cdots,\left\{(1,a_1^k),(2,a_2^k),\cdots,(s,a_s^k)\right\}\end{Bmatrix}\Bigg|_{k,s,a_i\in N}\right)\equiv$$
$$\equiv\begin{Bmatrix}\left\{(1,a_2^1),\cdots,(s-1,a_s^1)\right\},\cdots \\ \cdots,\left\{(1,a_2^k),\cdots,(s-1,a_s^k)\right\}\end{Bmatrix}\Bigg|_{k,s,a_i\in N}$$

– видалення першого «стовпчика» реляції;

$$S_R : S_R\left(\left\{\begin{matrix}\left\{\left(1,a_1^1\right),\left(2,a_2^1\right),\cdots,\left(s,a_s^1\right)\right\},\cdots\\ \cdots,\left\{\left(1,a_1^k\right),\left(2,a_2^k\right),\cdots,\left(s,a_s^k\right)\right\}\end{matrix}\right\}\Big|_{k,s,a_i\in N}\right)\equiv$$

$$\equiv\left\{\begin{matrix}\left\{\left(1,a_1^1+1\right),\left(2,a_2^1\right),\cdots,\left(s,a_s^1\right)\right\},\cdots\\ \cdots,\left\{\left(1,a_1^k+1\right),\left(2,a_2^k\right),\cdots,\left(s,a_s^k\right)\right\}\end{matrix}\right\}\Big|_{k,s,a_i\in N}$$

– збільшення першого «стовпчика» реляції на 1;

$$C_R : C_R\left(\begin{matrix}\left\{\left\{\left(1,a_1^1\right),\left(2,a_2^1\right),\cdots,\left(s,a_s^1\right)\right\},\cdots\\ \cdots,\left\{\left(1,a_1^k\right),\left(2,a_2^k\right),\cdots,\left(s,a_s^k\right)\right\}\right\}\Big|_{k,s,a_i\in N}\end{matrix}\right)\equiv$$

$$\equiv\left\{\left\{\left(1,0\right)\right\}\right\}$$

– встановлення константи $\left\{\left\{\left(1,0\right)\right\}\right\}$;

$$\circ_R : \circ_R\left(\left\{A^1,\cdots A^k\right\},\left\{B^1,\cdots B^s\right\}\right)\Big|_{\substack{k,s,m,n\in N,\,A^i\equiv\left\{\left(1,a_1^i\right),\left(2,a_2^i\right),\cdots\left(n,a_n^i\right)\right\},\\ Bi\equiv\left\{\left(1,b_1^j\right),\left(2,b_2^j\right),\cdots\left(m,a_m^j\right),\right\}}}\equiv$$

$$\equiv\left\{\begin{matrix}\circ\left(A^p,B^q\right)\Big|_{p=\overline{1,k},q=\overline{1,s}};\\ \left(\circ\left(A^p,B^q\right)=\left\{\begin{matrix}\left(1,a_1^p\right),\cdots\\ \cdots,\left(n,a_n^p\right),\left(n+1,b_1^q\right),\cdots,\left(m+1,b_m^q\right)\end{matrix}\right\}\right)\end{matrix}\right\}$$

– розширений декартовий добуток двох реляцій;

$$I_m^n : I_m^n\left(R_1,\cdots,R_n\right)=R_m\,\Big|_{m,n\in N,m\le n}$$

де $R_i\in R$;

– вибір реляції R_m;

$$=_R:=_R\left(R_1,R_2\right)=\begin{cases}T,\ \textit{якщо}\ R_1=R_2\\ F,\ \textit{інакше}\end{cases}$$

– предикат рівності реляцій.

Безпосередньо з наведеного виливає, що реляційні функції $\Pi_R,S_R,C_{\{0\}},\circ_R$ та реляційний предикат $=_R$ є реляційними моделями відповідних кортежних функцій $\Pi,S,C_0,\circ$ та кортежного предикату $=_K$ [15; 16]. Адже легко пересвідчитись, що:

$$\Pi_R\left(k\right)\big|_{k\in K}\equiv=\Pi\cdot H\left(k\right),$$

$$H\cdot S_R\left(k\right)\big|_{k\in K}\equiv S\cdot H\left(k\right),$$

$$H\cdot C_{\{0\}}\left(k\right)\big|_{k\in K}\equiv C_0\cdot H\left(k\right),$$

$$H\cdot\circ_R\left(k_1,k_2\right)\big|_{k_1,k_2\in K}\equiv\circ\cdot H\left(k\right),$$

$$H\cdot=_R\left(k_1,k_2\right)\big|_{k_1,k_2\in K}\equiv=_K\cdot H\left(k\right)$$

Звідси безпосередньо випливає справедливість наступного твердження.

Теорема

$\sigma_R=\left\{\Pi_R,S_R,C_{\{0\}},\circ_R,\psi,\chi,=_R,I_m^n\right\}\big|_{m,n\in N;m\le n}$ – I_m^n-базис алгебри ППА $A_K^{\text{чр}}$.

Дане твердження представляє алгебраїчну характеристику класу частково-рекурсивних функцій та частково-рекурсивних предикатів над реляціями як платформу семантико-синтаксичних специфікацій класу реляційних перетворень.

Висновки. Відомо, що рішення будь-якої задачі суть інтеграція рішень її підзадач. Для відносно простих задач інтеграційний аспект частіше за все не є визначальним і їм можна знехтувати. При вирішенні ж реальних задач інтеграційний аспект відіграє ключову роль. Така особливість є характерною для задач побудови прикладних інформаційно-телекомунікаційних систем. Тому дослідження загально значимих інтеграційних структур для вирішення реальних задач, в області прикладних телекомунікацій, набувають сьогодні першорядного значення. Особливістю таких задач сьогодні є залучення до їх вирішення великих масивів складно організованої інформації. Строга специфікація складних інформаційних структур має першорядне значення як власне для визначення використовуваних структурованих типів даних, так і для дослідження різного роду маніпуляцій над ними.

У статті на базі поняття іменної множини розглянуто метод іменних специфікацій складно структурованих даних. Запропоновано іменні специфікації даних типу кортежу та реляції (скінченого відношення). Представлено та обґрунтовано метод ізоморфних специфікацій знаходження породжуючих сукупностей у примітивних програмних алгебрах обчислюваних функцій та предикатів над зліченими носіями. Вирішено проблему повноти для класу реляційних *чр*-функцій та *чр*-предикатів. Показано, що сигнатурні операції ППА складають повну систему операцій, що уточнюють поширені методи побудови прикладних телекомунікаційних систем. Отримані результати розвивають напрямок прикладних телекомунікаційних систем програмування, що досліджує електронні комунікації, перед усім, як інструмент, що підтримує вирішення реальних прикладних задач.

Список літератури:

1. Redko V. Foundations of programmology. *Cybernetics and Systems Analysis*. 2000. № 36. P. 27–42. DOI:10.1007/BF02733300.
2. Редько І.В., Редько Д.І., Захарченко Т. Л. Концептологічні основи проектування. Київ : Компринт. 2016.
3. Редько І.В., Яганов П.О., Зилевіч М.О., Редукційне концептування оракульних схем. *Системні дослідження та інформаційні технології*, 2021. № 1. С. 21–33, DOI: 10.20535/SRIT.2308-8893.2021.1.02
4. Редько І.В., Яганов П.О., Концептуальна модель технологічного середовища програмування. *Наукові вісті КПІ.* 2020. № 1. С. 18–26. DOI: 10.20535/kpi-sn.2020.1.197953.
5. Bacus, J.W. The algebra of functional programs: Function level reasoning, linear equations, and extended definitions. In *Lecture Notes in Computer Science*, № 107. *Formalization of Programming Concepts*. Springer-Verlag, New York, 1981, C. 1–43.
6. Curry H.D., Hindley R., Seldin J.P. Combinatory Logic. Studies in Logic. Amsterdam: North-Holland Co. 1972.
7. Dahl O.-J., Dijkstra E.W., Hoare C.A.R. Structured Programming. London: Academic Press,. 1972. ISBN 0-12-200550-3.
8. McCarthy J. Recursive Functions of Symbolic Expressions and Their Computation by Machine, *Communications of the ACM*. 1960. № 3. P. 184–195.
9. ACM Turing Award Lectures: The First Twenty Years: 1966 to 1985. Assn for Computing Machinery. 1987.
10. Church A. Introduction to Mathematical Logic. Princeton, New Jersey: Princeton University Press, 1956/
11. Dijkstra E.W. A Discipline of Programming. Prentice-Hall. 1976.
12. Wirth N. Algorithms and Data Structures. Oberon. 2004.
13. Brooks, F.P.Jr. Design of Design, The: Essays from a Computer Scientist. New York: Addison-Wesley. 2010.
14. Comella-Dorda S., Wallnau K., Seacord R., Robert J. A Survey of Legacy System Modernization Approaches. Pittsburgh: Software Engineering Institute. 2000.
15. Redko D.I., Redko I.V., Yahanov P. O., Zakharchenko T.L. Compositional basis in programmer activity. *Системні дослідження та інформаційні технології*. 2015. № 4. P. 83–96.
16. Захарченко Т.Л., Редько І.В., Д. І. Редько, Яганов П.О. Примітивна програмна алгебра обчислюваних функцій над записами. *Наукові вісті НТУУ «КПІ».* 2015. № 2. С. 29–40.
17. Захарченко Т.Л., Редько І.В. Проблема повноти в класі функцій над записами, які зберігають денотати. *Наукові вісті НТУУ «КПІ».* 2015. № 5. С. 23–30.
18. Cutland N. Computability. An introduction to recursive function theory. Cambridge university press. 1980.
19. Goncharov S. Computable single-valued numerations. Springer. *Translated from Algebra i Logika*. V. 19. № 5. 1980. P. 507–551. URL: https://link.springer.com/article/10.1007/BF01669607

Redko I.V., Sushchenko V.S. COMPLETENESS ISSUES IN THE CLASSES OF COMPUTABLE FUNCTIONS AND PREDICATES OVER NAMED DATA STRUCTURES

The article examines the algebraic properties of classes of computable functions and predicates over named data structures. Many studies justify their role as a universal means of specifying pragmatically important data structures for programming. The main focus is on the compositional programming paradigm as a methodological basis for considering methods of program construction. Primitive program algebras (PPA) of partial-recursive functions and partial-recursive predicates over named data represent the object of research. The selection of these data structures is determined by their importance and popularity in both theoretical and applied programming. In particular, tuple and relational data structures play a crucial role in research related to the completeness of query languages in databases. The definitions of PPA are provided in the works of V.N. Redka, D.B. Buia, I.V. Redka, and other authors. The PPA signature consists of superposition, branching, and looping operations, which refine the well-known control structures of most programming languages and can be applied in telecommunication software-hardware system development. Significant attention is given to the challenges of constructing PPA characteristics for classes of considered computable functions and predicates, which are the article's subject.

The computability of functions (predicates) over effectively enumerable domains is introduced as numeral computability, in the sense of works by N. Cutland in "Computability: An Introduction to Recursive Function Theory," A. Ershov in "Computability in Arbitrary Domains and Bases," and Y. Ershov in "The Theory of Numerations." The present study is dedicated to applying the algebraic approach, particularly the method of isomorphic specifications in research and software development. Along with general results regarding named data and functions, the paper describes classes of tuple and relational computable functions and predicates. Corresponding basic sets of functions and predicates in PPAs are defined.

Key words: *algebraic characteristic, compositional paradigm, computable function, computable predicate, tuple, relation, program algebra, isomorphism, Imn-basis.*